Chapter 03

Networking and Security

Mr. Nilesh Vishwasrao Patil
Government Polytechnic Ahmednagar

Socket

- Network socket is an endpoint of an inter-process communication flow across a computer **network**.
- Sockets provide the communication mechanism between two computers using TCP/IP.

Socket

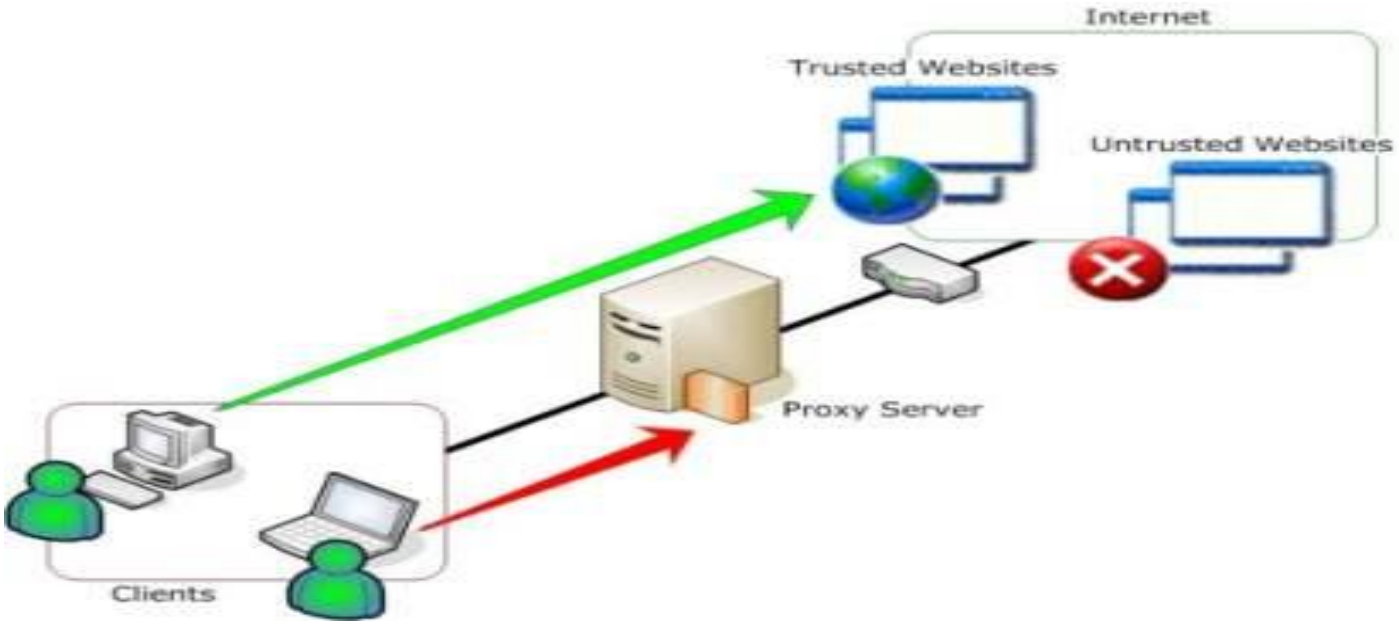
- IP (Internet Protocol): Low level routing protocol. Divides data into small packets and send on given address. It does not guarantee to deliver the packets.
- Transmission Control Protocol (TCP): Higher level protocol. Reliability to deliver data.
- User Datagram Protocol (UDP): Next to TCP. It support fast, connectionless, unreliable transport of data packets.

Difference Between TCP & UDP

TCP	UDP
Connection oriented	Connectionless
Reliable	Unreliable
Retransmit	No retransmission
Slower data transmission	Faster data transmission
Require most cost	Less cost than TCP

Proxy Server

- It is mediator between real web server and a client applications like web browser.
- Filter specific request and stored data in catch for future use.



Reserved Sockets Ports

- Port number range : 0 to 65535
- 1 to 1024 are reserved.
- Examples:
  - FTP : 21
  - Telnet : 23
  - Email : 25
  - HTTP: 80
- Client request for file from HTTP server, it is called hits.

Internet Addressing

- Internet address is unique number, used to identify each machine uniquely.
- IP address: 2 version
  - IPv4 : 32-bits and in decimal (Now)
  - IPv6 : 128-bits and in hexadecimal (Future)
- IPv4 : Divide 32 bits in 4 parts.
- Each part range from 0 to 255.

Internet Addressing

- Divided into 5 classes:
  - Class A
  - Class B
  - Class C
  - Class D
  - Class E

Assignment

- DNS (Domain Name Services)
- Internet
- Server – Client
- Relationship between Java and Internet
- Web server and Application server with one example at least.

Socket Programming

- A client program creates a socket on its end of the communication and attempts to connect that socket to a server.
- When the connection is made, the server creates a socket object on its end of the communication.
- The client and server can now communicate by writing to and reading from the socket.

Java Sockets Programming

- The package java.net provides support for sockets programming.

- Typically you import everything defined in this package with:

```
import java.net.*;
```

Classes

InetAddress

Socket

URL

URLConnection

ServerSocket

DatagramSocket

DatagramPacket

InetAddress class

- InetAddress class is encapsulate both numeric IP address (eg .74.125.236.88) and the domain name (eg. www.google.com) for the address.
- Interaction with this class by using the Hostname rather than IP address, more conveniently and understandable way.
- For example, mostly every internet user don't know IP address for google.com.

InetAddress class

- It has both Factory and Instance methods:
- Factory method:
  - is a static method in a class return an instance of that class.
- Instance Methods:
  - is a non-static method.

About InetAddress class

- As we know, **"new"** Keyword is used to create object to that corresponding class.
- InetAddress Class has no visible constructors. to create a InetAddress object.
- **Factory Method** is used to create objects.
- Three factory methods:
  - static InetAddress getLocalHost()
  - static InetAddress getByName(String hostName)
  - static InetAddress[] getAllByName(String hostName).
  - All methods generate : UnknownHostException

Instance Methods

- boolean equals(Object other)
- **byte[] getAddress(): Return four element of IP address.**
- **String getHostAddress(): Return host address associated with InetAddress.**
- **String getHostName(): Return host name.**
- **int hasCode() : return hashcode of invoking object.**
- **Boolean isMultiCastAddress()**

URL

- URL is Uniform Resource Locator.
- It is a formatted string used by email clients, web browsers and different type of software recognize network resource on the internet.
- Network resource could be text, documents, plain web pages, programs or graphics.

URL

- URL string consist of following parts:
  - Network protocol
  - Host name or address
  - Port number
  - File or resource location.
- URL provides comprehensive form to uniquely identify or address information on the internet.
- Java has provided : **URL class**

URL

- URL string consist of three parts:
  - Network protocol
  - Host name or address
  - File or resource location.
- URL provides comprehensive form to uniquely identify or address information on the internet.
- Java has provided : **URL class**

URL

- Ex
  - http://www.msbte.com/index.html
- URL class has some constructors and it throws MalformedURLException
- URL(String url)
- URL(String protocol, String hostname, int port, String path)
- URL(URL obj, String url)

URL

- String getProtocol()
- String getHost()
- String toExternalForm()
- String getFile()
- String getPort()

URLConnection Class

- Used for accessing the attributes of remote resource.

- public URLConnection openConnection()throws IOException{}
  - openConnection() of URL class returns the object of URLConnection class.

URLConnection Class methods

- int getContentLength() : Return size of contents related to resource. If no length then return -1.
- String getContentType(): Return type of content of resource.
- long getDate() : Return date and time of response
- long getLastModified() : return last date and time modified of response

URLConnection Class methods

- Long getExpiration(): Return expiration date and time in miliseconds.
- InputStream getInputStream() : Used to get contens of resource.

Socket Programming

- Sockets provide the communication mechanism between two computers using TCP.
- A client program creates a socket on its end of the communication and attempts to connect that socket to a server.
- When the connection is made, the server creates a socket object on its end of the communication.
- The client and server can now communicate by writing to and reading from the socket.

Socket Programming

- Socket class represents a socket.
- ServerSocket class provides a mechanism for the server program to listen for clients and establish connections with them.

Steps to establish connection

- The server instantiates a ServerSocket object, denoting which port number communication is to occur on.
- The server invokes the accept() method of the ServerSocket class. This method waits until a client connects to the server on the given port.
- After the server is waiting, a client instantiates a Socket object, specifying the server name and port number to connect to.

Steps to establish connection

- The constructor of the Socket class attempts to connect the client to the specified server and port number. If communication is established, the client now has a Socket object capable of communicating with the server.
- On the server side, the accept() method returns a reference to a new socket on the server that is connected to the client's socket.

Steps to establish connection

- Each socket has both an OutputStream and an InputStream.
- The client's OutputStream is connected to the server's InputStream,
- Client's InputStream is connected to the server's OutputStream.

ServerSocket Constructor

- public ServerSocket(int port)
- public ServerSocket(int port, int backlog)
- public ServerSocket(int port, int backlog, InetAddress address)
- public ServerSocket()

ServerSocket Methods

- public int getLocalPort() : Return port number of server socket is listening.
- public Socket accept() : Waits for an incoming client.
- public void setSoTimeout(int timeout) : Sets the time-out value for how long the server socket waits for a client during the accept().
- public void bind(SocketAddress host, int backlog) : Binds the socket to the specified server and port in the SocketAddress object. Use this method if you instantiated the ServerSocket using the no-argument constructor.

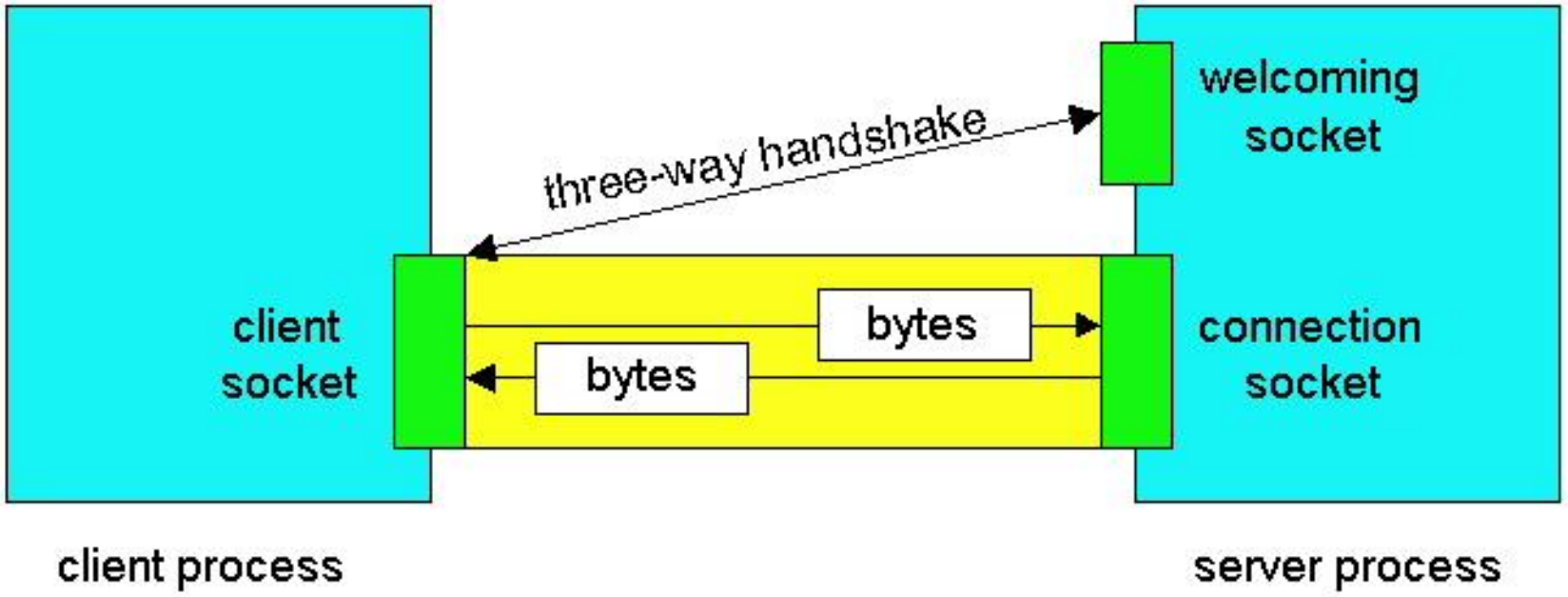
Socket Constructor

- public Socket(String host, int port)
- public Socket(InetAddress host, int port).
- public Socket(String host, int port, InetAddress localAddress, int localPort)
- public Socket(InetAddress host, int port, InetAddress localAddress, int localPort)
- public Socket()

Socket Methods

- public void connect(SocketAddress host, int timeout)
- public InetAddress getInetAddress()
- public int getPort()
- public int getLocalPort()
- public SocketAddress getRemoteSocketAddress()
- public InputStream getInputStream()
- public OutputStream getOutputStream()
- public void close()

Sockets



Client socket, welcoming socket (passive) and connection socket (active)

Client/server socket interaction: TCP